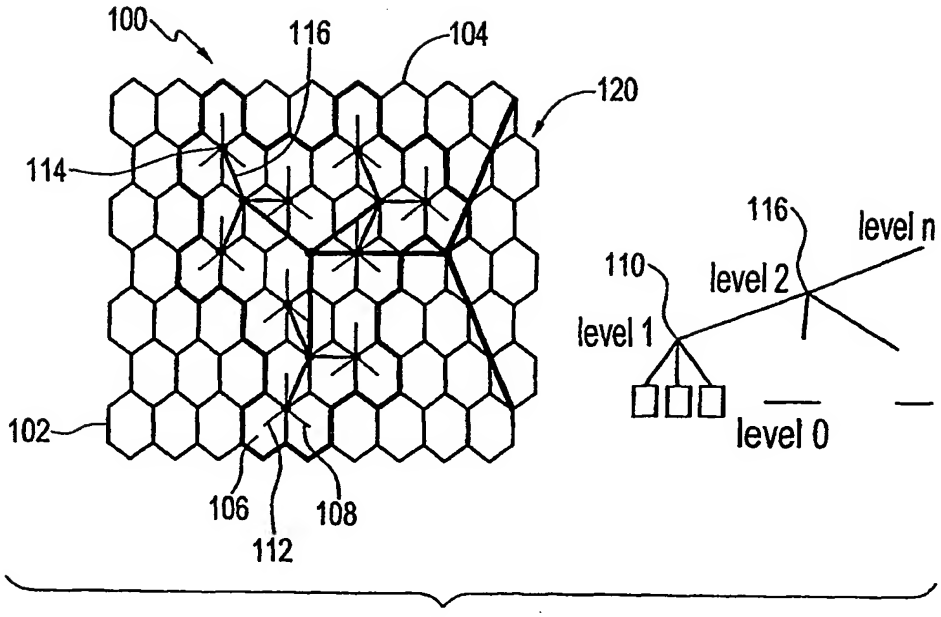
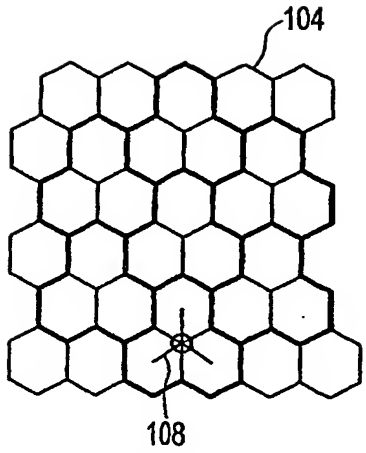
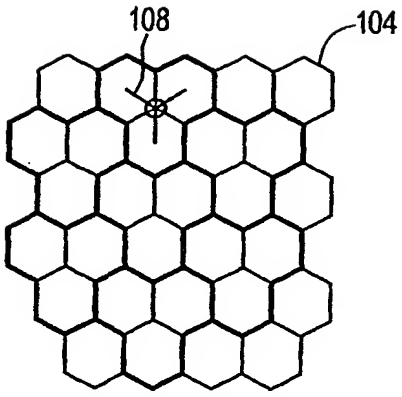

100
116
104
120
114
102
106
112
108
116
level n
110
level 2
level 1
level 0

FIG. 1



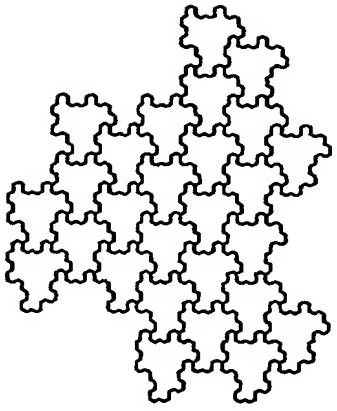
(a) ⊥ Type

FIG. 2A



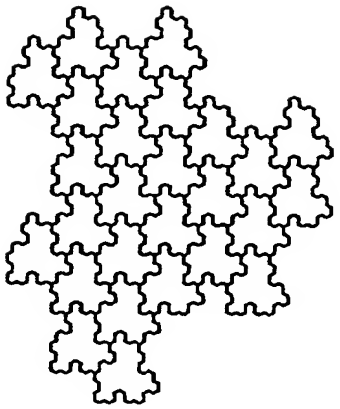
(b) Y Type

FIG. 2B



(a)

FIG. 3A



(b)

FIG. 3B

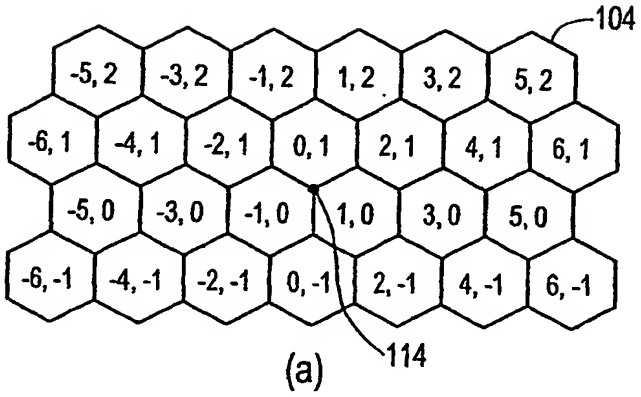

104
-5, 2
-3, 2
-1, 2
1, 2
3, 2
5, 2
-6, 1
-4, 1
-2, 1
0, 1
2, 1
4, 1
6, 1
-5, 0
-3, 0
-1, 0
1, 0
3, 0
5, 0
-6, -1
-4, -1
-2, -1
0, -1
2, -1
4, -1
6, -1
114
(a)

FIG. 4A

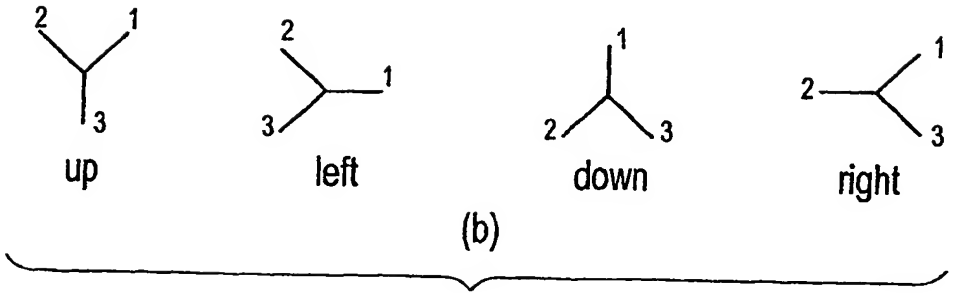

2
1
3
up
2
1
3
left
1
2
3
down
1
2
3
right
(b)

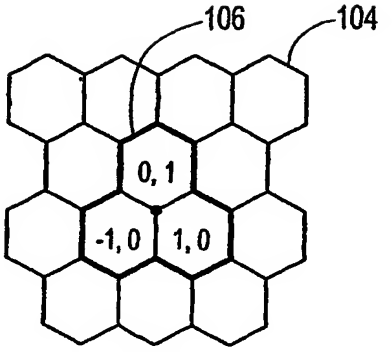
FIG. 4B

```
Setup_Y_tree(C)
{
 sub-tree1 = Create_leaf(0,1);
 sub tree2 = Create_leaf(-1,0);
 sub-tree3 = Create_leaf(1,0);
 Tree=Compose_tree(sub-tree1,sub-tree2,
   sub-tree3,C(i));
 z = 2/3; x = 1, y = 1/3;
 for i=2:n
  {
   if (i is even)
    { z = z * 3; y = y * 3; }
   else
    x = x * 3;

   case C(i)
    "up":
     { x1= x; x2= -x; x3=  0;
       y1= y; y2=  y; y3= -z; }
    "left":
     { x1= z; x2= -x; x3= -x;
       y1= 0; y2=  y; y3= -y; }
    "down":
     { x1= 0; x2= -x; x3=  x;
       y1= z; y2= -y; y3= -y; }
    "right":
     { x1= x; x2= -z; x3=  x;
       y1= y; y2=  0; y3= -y; }

 Copy(sub-tree1,Tree);
 Copy(sub-tree2,Tree);
 Copy(sub-tree3,Tree);
/* Copy Tree to sub-trees */
 Shift(sub-tree1, x1, y1);
```

FIG. 5A

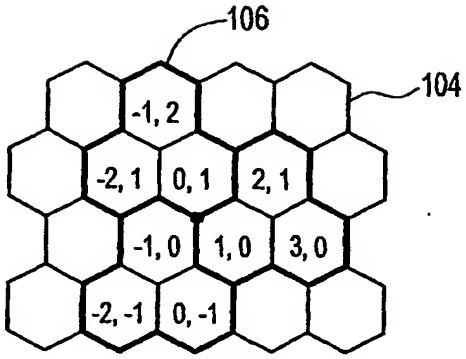
```
    Shift(sub-tree2, x2, y2);
    Shift(sub-tree3, x3, y3);
   /* Shift the coordinates of every leaf in
      Tree by x(k) and y(k) respectively */
    Tree=Compose_tree(sub-tree1,sub-tree2,
      sub-tree3,C(i));
   }
 }

 Create_leaf(x,y)
 {
  Create_tree_node(leaf);
  leaf->x = x;
  leaf->y = y;
  return(leaf);
 }

 Compose_tree(sub-tree1,sub-tree2,sub-tree3,
      orientation)
 {
  Create_tree_node(new_root);
  new_root -> child1 = sub-tree1;
  new_root -> child2 = sub-tree2;
  new_root -> child3 = sub-tree3;
  new_root -> orientation = orientation;
  return(new_root);
 }
```

FIG. 5B



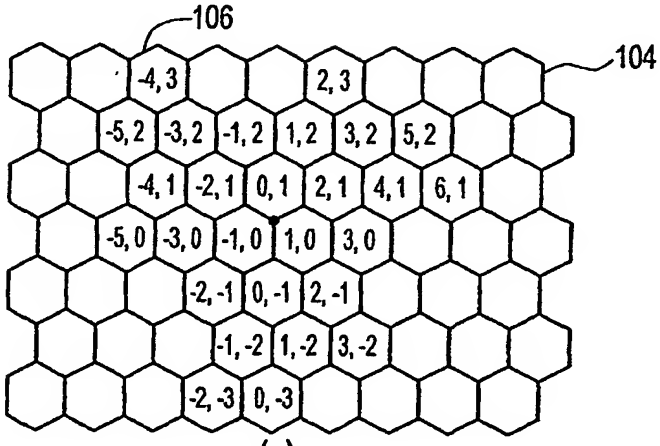
(a)

FIG. 6A



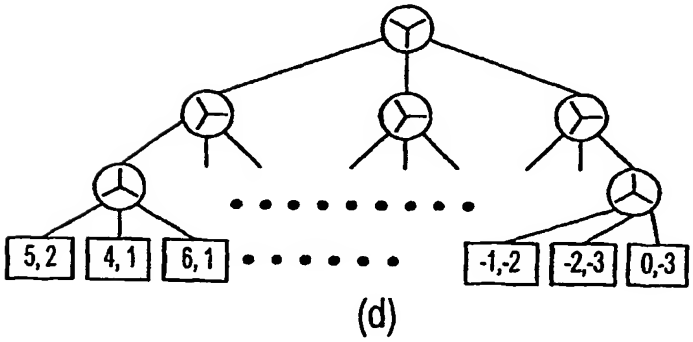
(b)

FIG. 6B



(c)

FIG. 6C



(d)

FIG. 6D

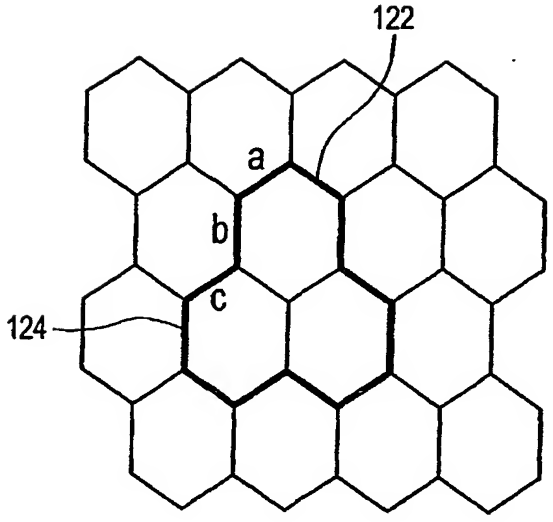


FIG. 7

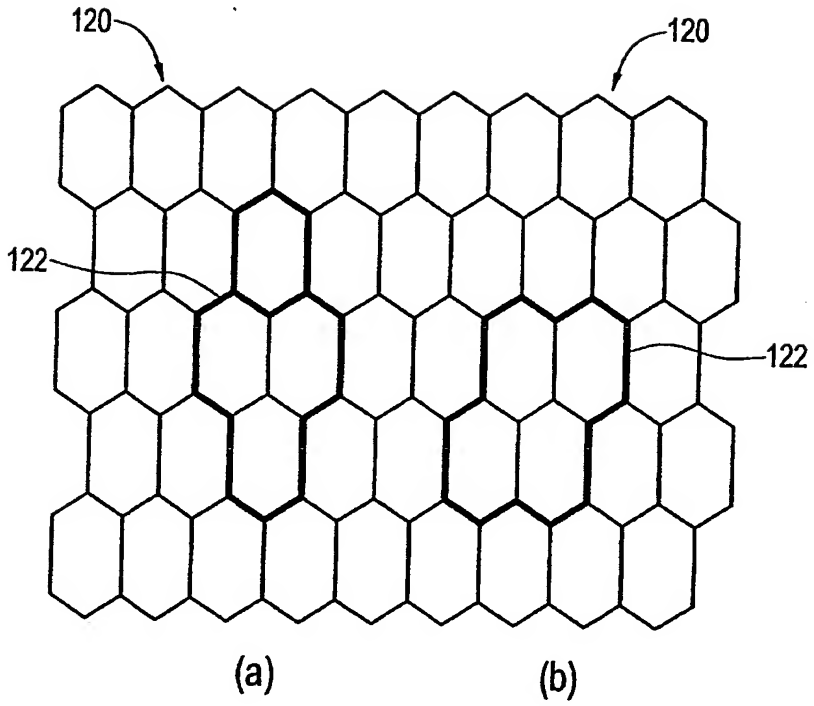


FIG. 8

```
Merge(A,B)
{
  /* Complement every bit in sequence B */
  BI = Bit_wise_complement(B);
  Bh = Reverse_bit_order(BI);
  do
    {
     /* Find the sub-sequences in A and Bh such
        that the two sub-sequences have the same
        pattern */
     if(Find_next_match(A,Bh,*(&sub) == false);
       break;

     /* Judge if the result satisfies the
        requirements */
     if(Accept(A,Bh,sub == true)
      {
         /* Rewrite sequence A and B */
         Rewrite sequence A = (A1)(sub)(A2);
         Rewrite sequence B = (B1)RevFlip(sub)(B2);

         /* Determine the sequences A12 and B12, which
            are the portions of A and B in the merged
            polygon C, respectively */
         Calculate A12 = ModMerge(A1, A2);
         Calculate B12 = ModMerge(B1, B2);

         /* Merge A12 and B12 and get C */
         C = (A12)(B12);
         Output(C);
      }
    }
}

RevFlip(Sub)
{
  /* Sub1 is the bit-wise complement of sequence Sub */
  Sub1 = Bit_wise_complement(Sub);
  /* Sub2 is the sequence of reversing the bit order
     of Sub1 */
  Sub2 = Reverse_bit_order(Sub1);
  Return Sub2;
}

ModMerge(S1, S2)
{
  /* S3 is the sequence of S2 followed by S1 */
  S3 = (S2)(s1);
  S4 = Complement_the_first_bit(S3);
  S5 = Delete_the_last_bit(S4);
  Return S5;
}
```

FIG. 9

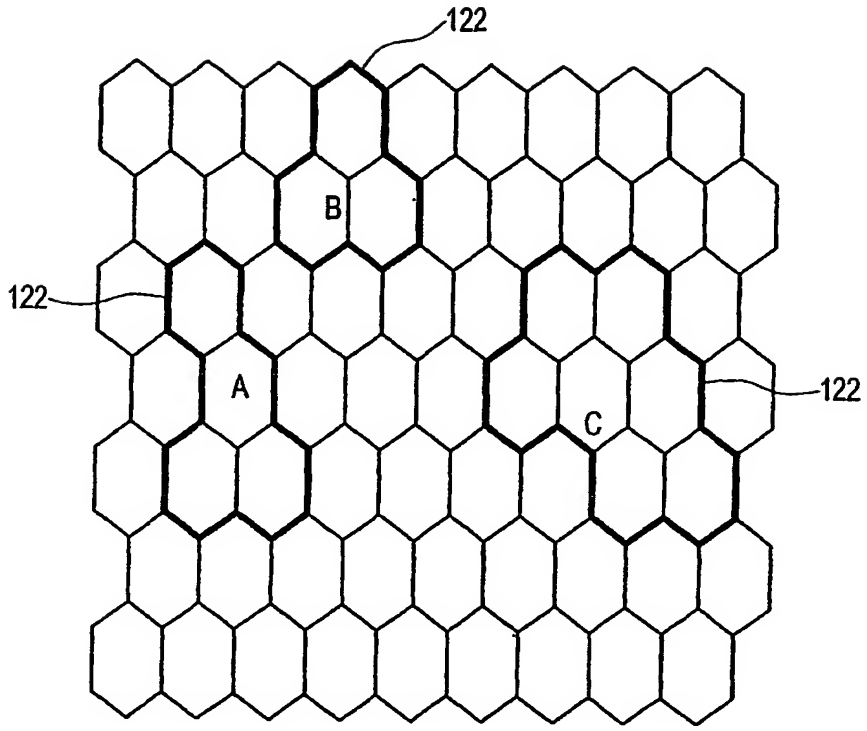


FIG. 10

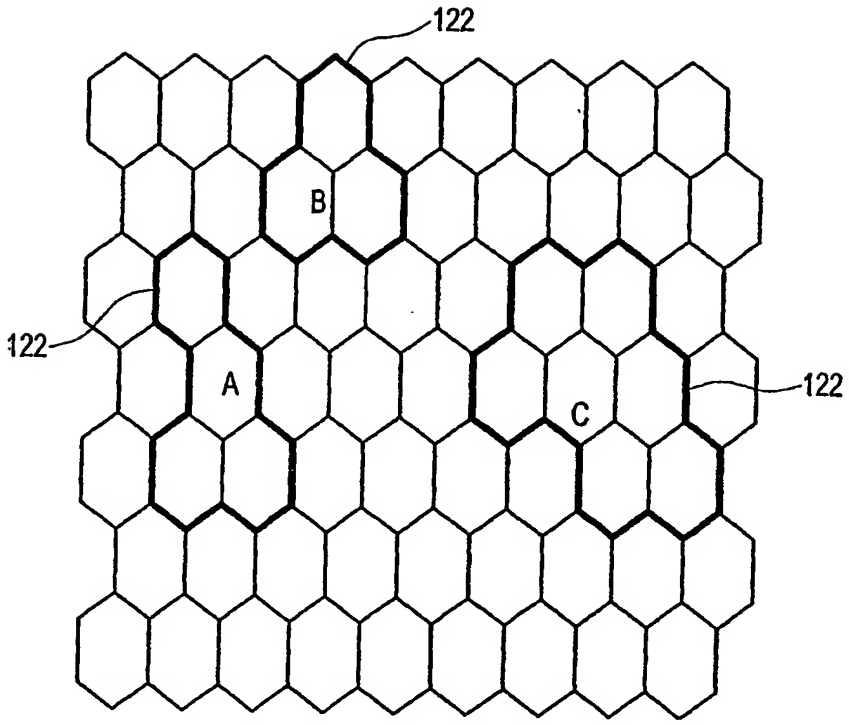


FIG. 11

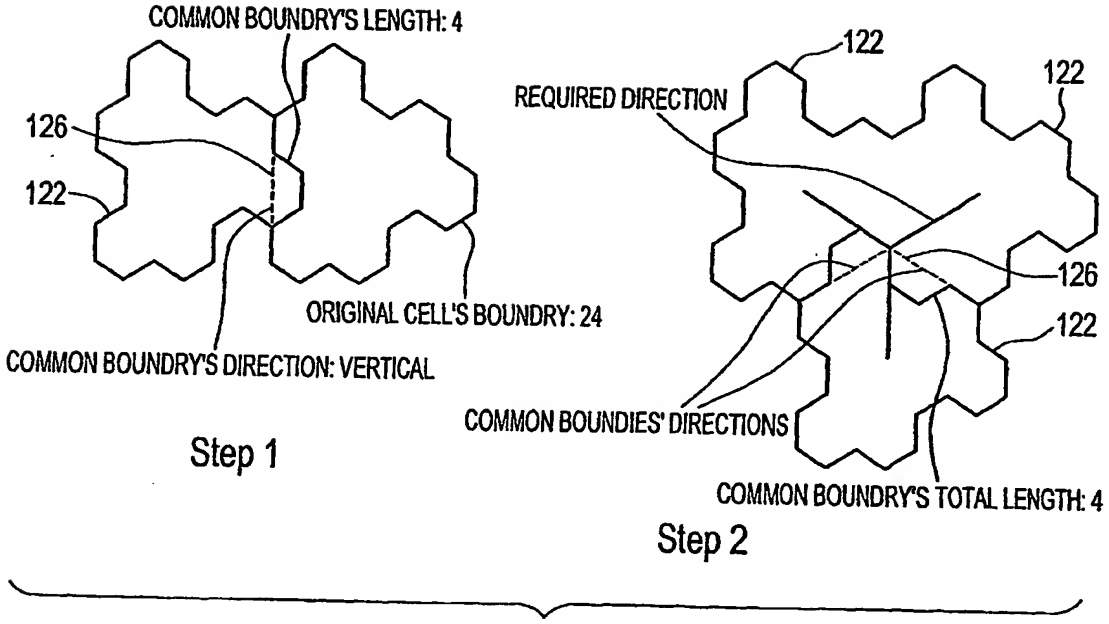

COMMON BOUNDRY'S LENGTH: 4
126
122
ORIGINAL CELL'S BOUNDRY: 24
COMMON BOUNDRY'S DIRECTION: VERTICAL
Step 1
122
122
REQUIRED DIRECTION
126
122
COMMON BOUNDIES' DIRECTIONS
COMMON BOUNDRY'S TOTAL LENGTH: 4
Step 2

FIG. 12

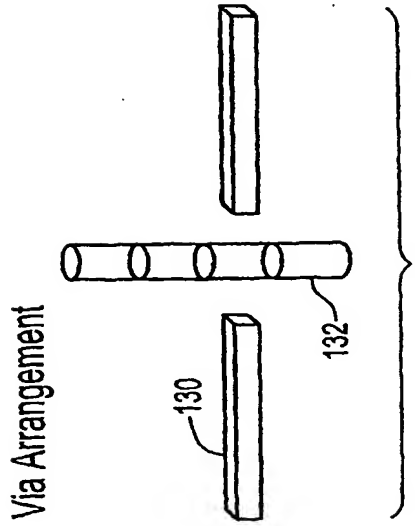


FIG. 13B

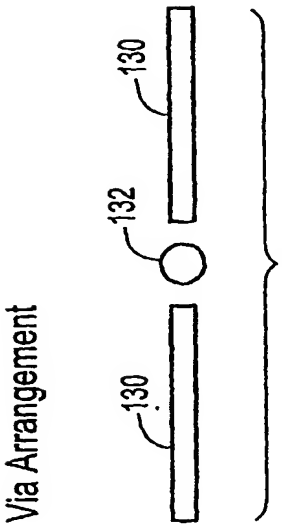


FIG. 13A

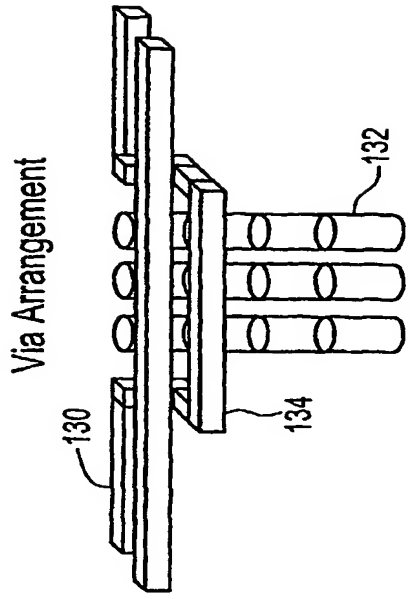


FIG. 14B

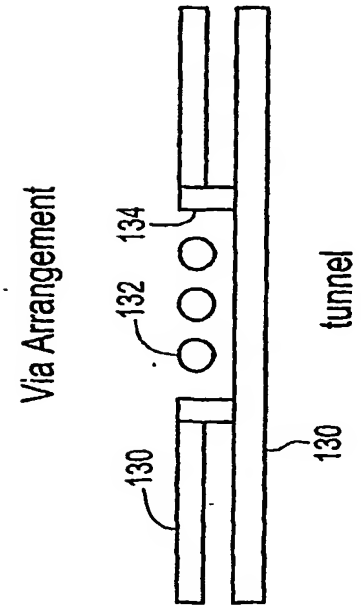


FIG. 14A

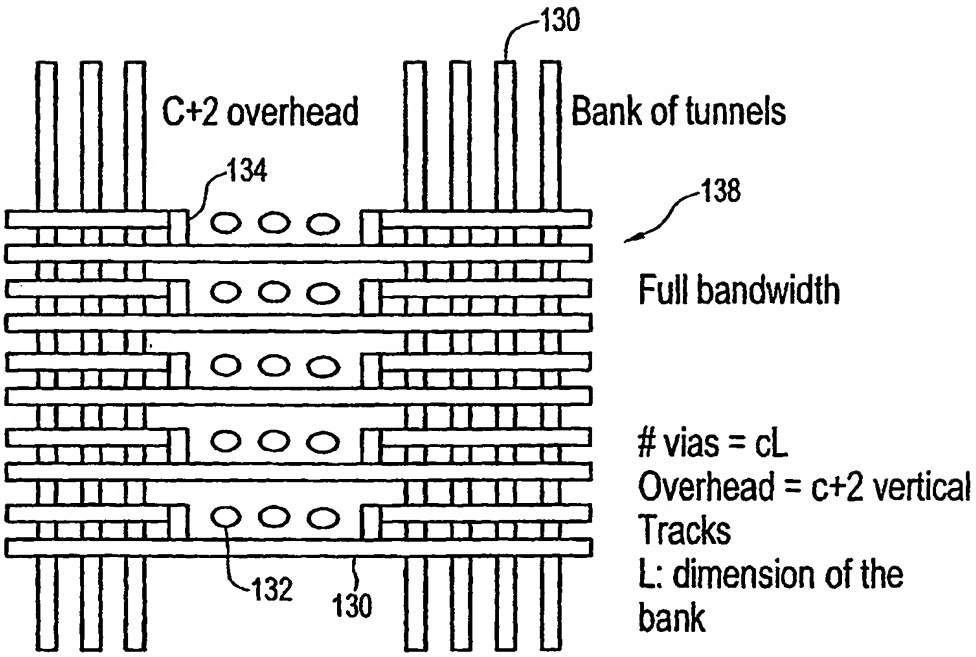

130
C+2 overhead
Bank of tunnels
134
138
Full bandwidth
# vias = cL
Overhead = c+2 vertical Tracks
L: dimension of the bank
132
130

FIG. 15

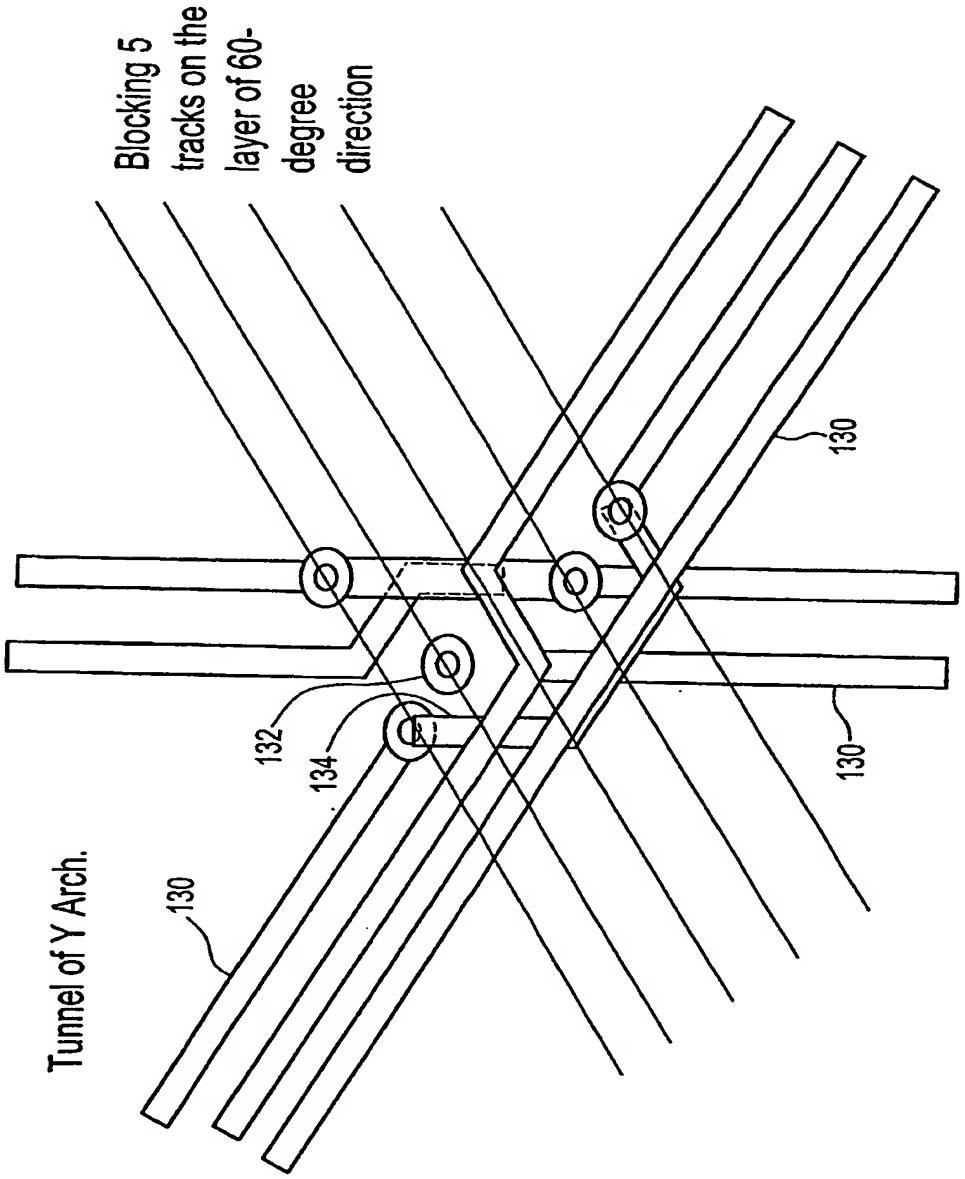

Blocking 5 tracks on the layer of 60-degree direction
Tunnel of Y Arch.
130
132
134
130
130

FIG. 16

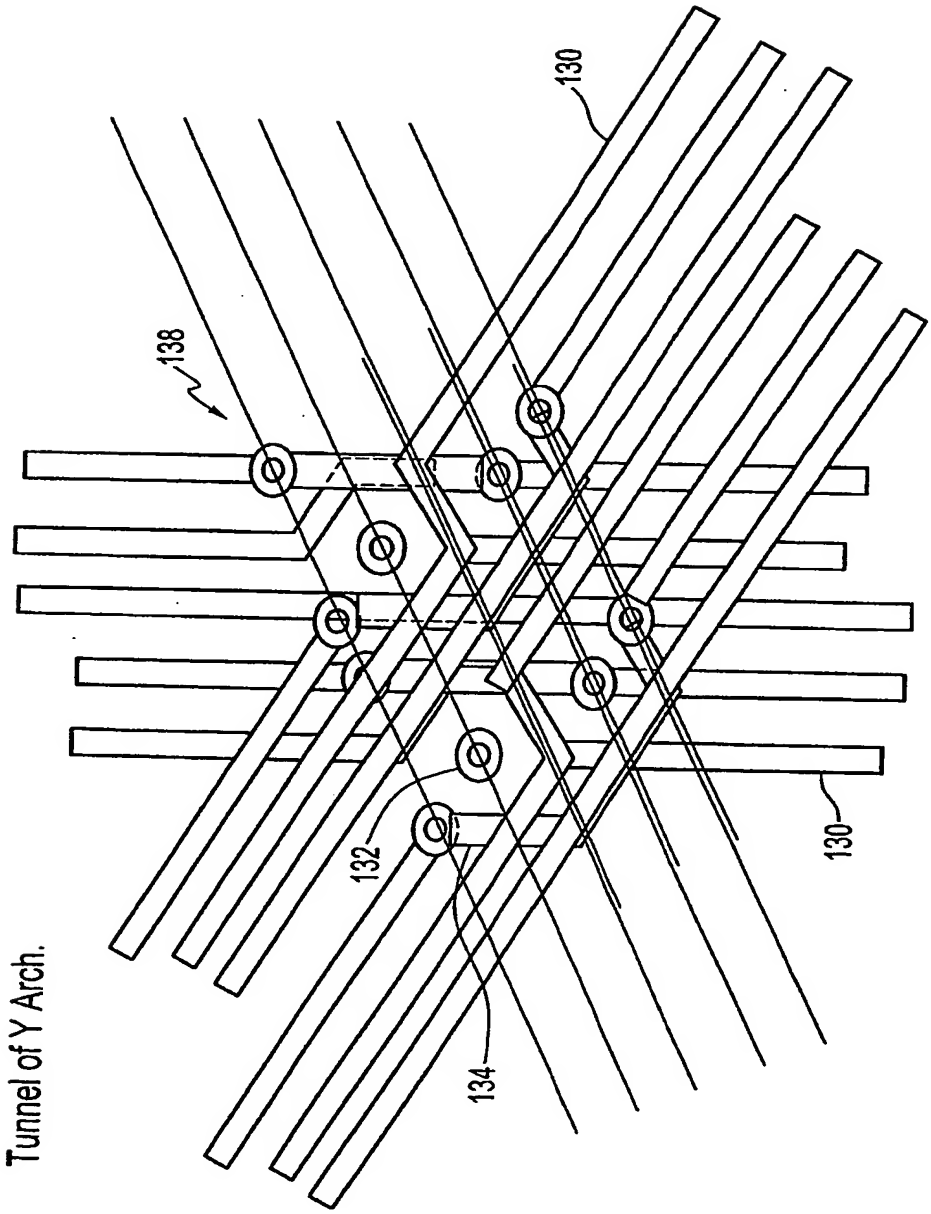


FIG. 17

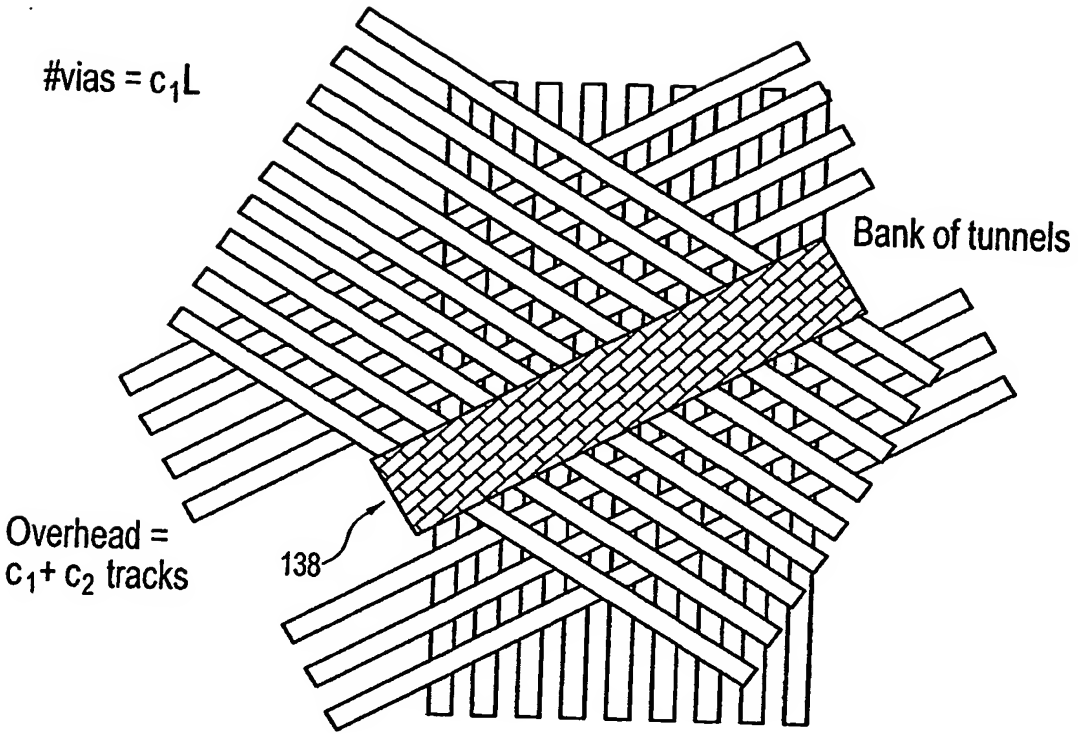


FIG. 18

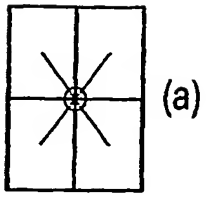


FIG. 19A

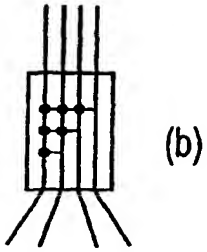


FIG. 19B

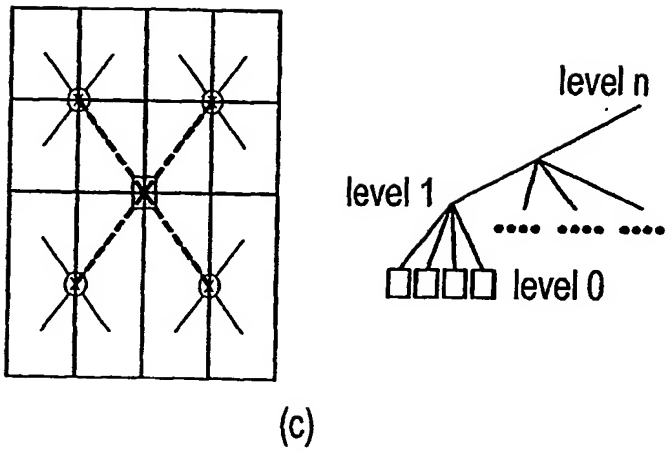


FIG. 19C

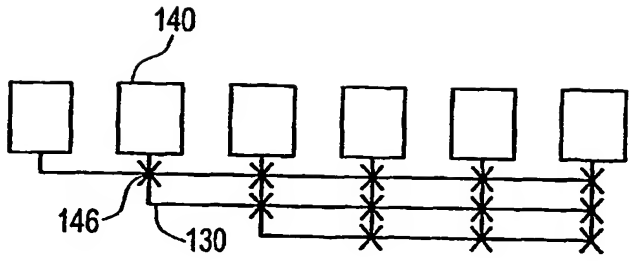
L_X	D_X
$L_1 = 2\sqrt{2}$ $L_n = 4L_{n-1} + 2^{3n-2}\sqrt{2}$	$D_1 = 6\sqrt{2}$ $D_n = 4D_{n-1} + 6 \cdot 2^{4n-4}\sqrt{2}(2^n - 1)$

FIG. 20

Recursive form solutions for L_Y and D_Y

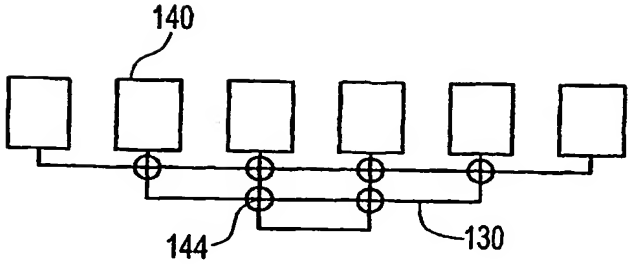
L_Y
$L_1 = \sqrt{3}\,a$ $L_n = 3L_{n-1} + 3^{\frac{3n}{2}-1}\,a$
D_Y
$D_1 = 2\sqrt{3}\,a$ $D_n = 3D_{n-1} + (3+\sqrt{3})(3^{\frac{n}{2}}-1)\,3^{2n-2}\,a$

FIG. 21



(a)

FIG. 22



(b)

FIG. 23

Model	L	D	M
a	$\frac{3n^2-2n}{8}$	$\frac{n(n+1)(n-1)}{6}$	$\frac{n^2(3n-2)(n-1)(n+1)}{48}$
b	$\frac{n^2}{4}$	$\frac{n(n+1)(n-1)}{6}$	$\frac{n^3(n-1)(n+1)}{24}$

FIG. 24

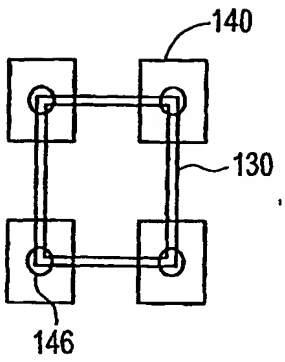


FIG. 25A

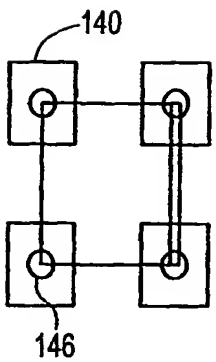


FIG. 25B

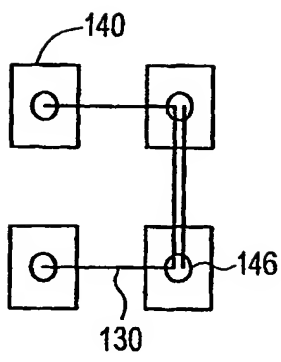


FIG. 25C

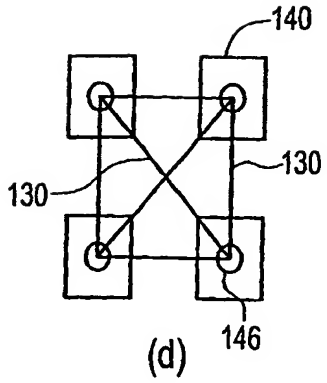


FIG. 25D

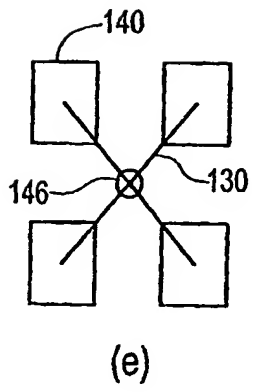


FIG. 25E

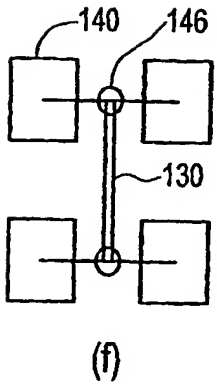


FIG. 25F

Model	L	D	M
a	8	8	64
b	5	8	40
c	4	10	40
d	$4+2\sqrt{2}$	$4+2\sqrt{2}$	$24+16\sqrt{2}$
e	$2\sqrt{2}$	$6\sqrt{2}$	24
f	4	10	40

FIG. 26

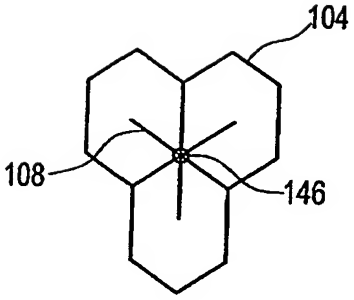


FIG. 27A

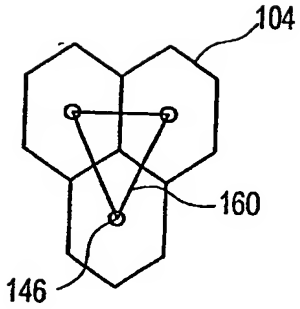


FIG. 27B

Model	L	D	M
Y	$\sqrt{3}$	$2\sqrt{3}$	6
Δ	3	3	9

FIG. 28

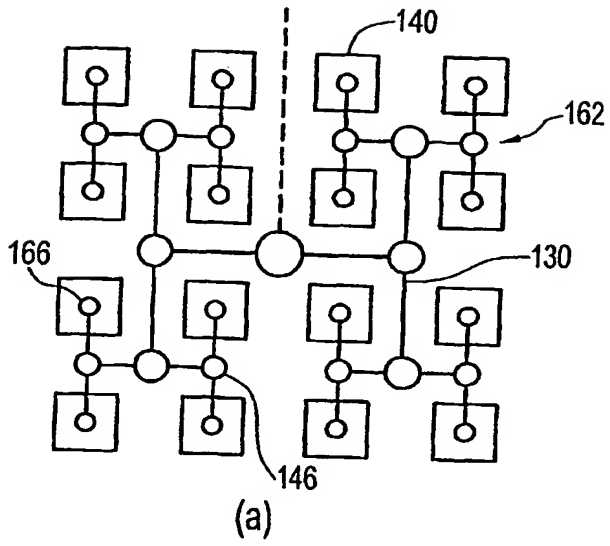


FIG. 29A

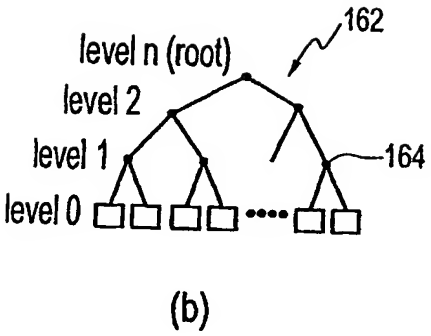


FIG. 29B

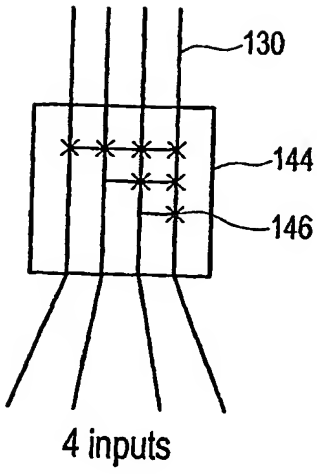


FIG. 30A

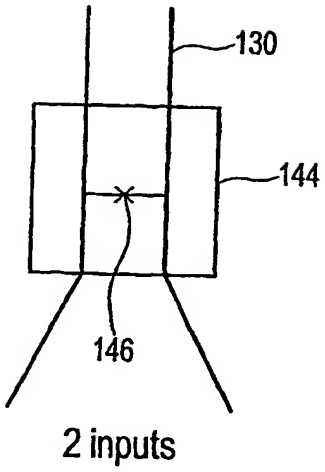


FIG. 30B

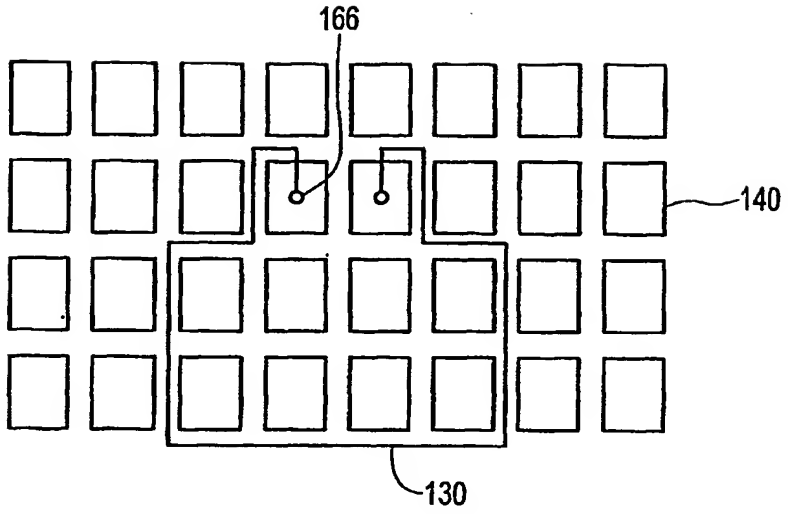


FIG. 31

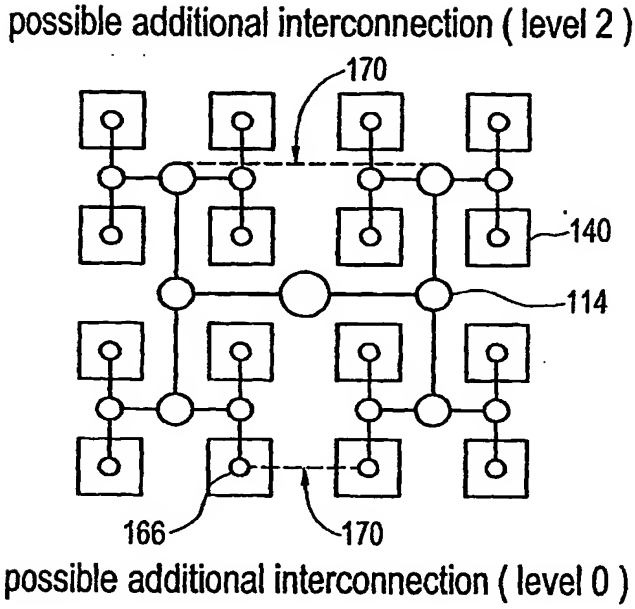

possible additional interconnection (level 2)
170
140
114
166
170
possible additional interconnection (level 0)

FIG. 32

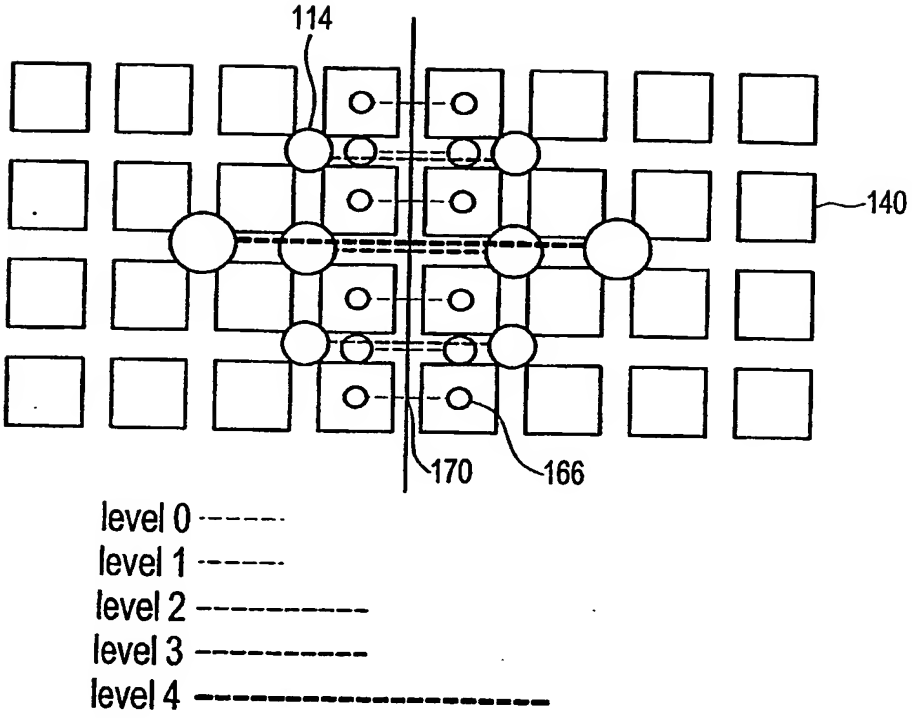


FIG. 33

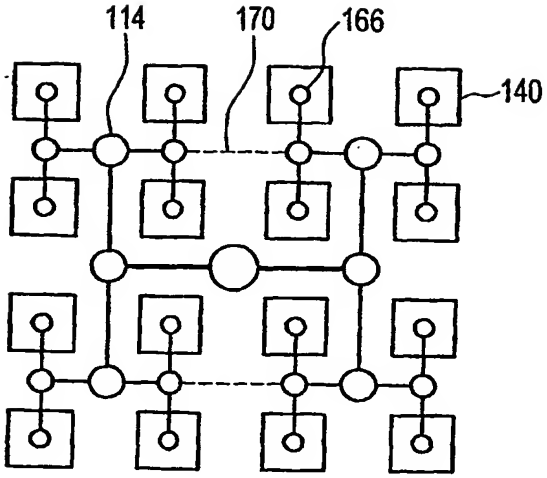
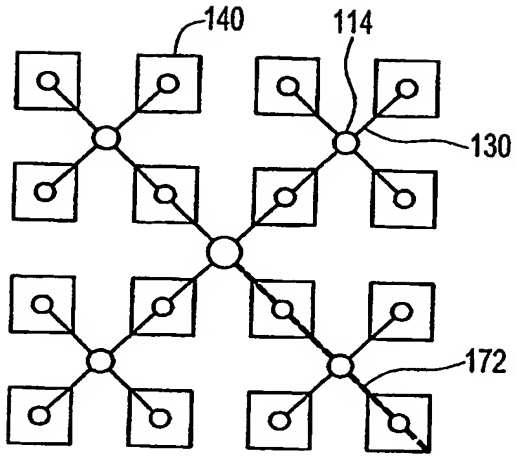


FIG. 34

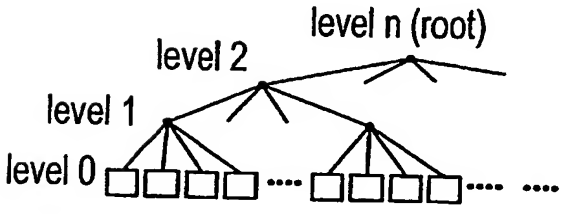
m	ΔL	ΔD	I
0	32	-1952	61
1	32	-3840	120
2	128	-14848	116
3	128	-28672	224
4	512	-106496	208
5	512	-196608	384
6	2048	-655360	320
7	2048	-1048576	512
8	8192	-2097152	256

FIG. 35



(a)

FIG. 36A



(b)

FIG. 36B

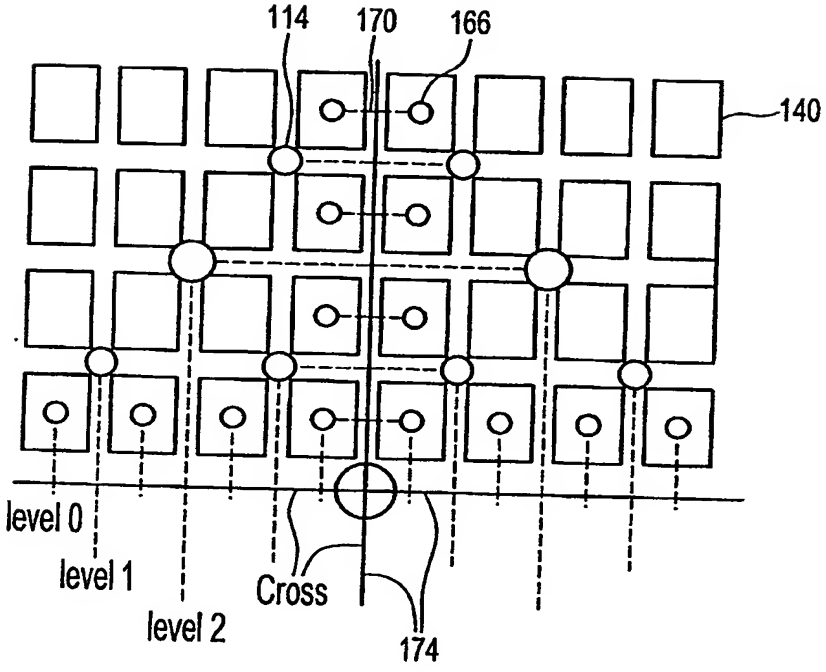


FIG. 37

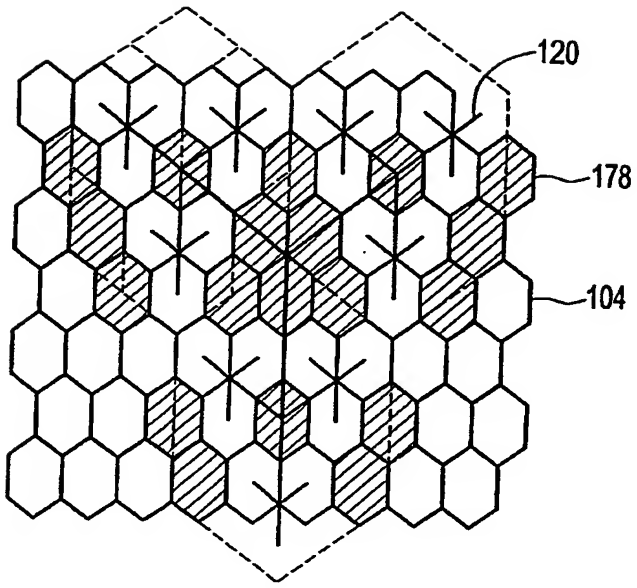


FIG. 38A

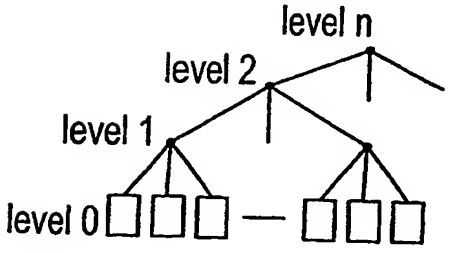


FIG. 38B

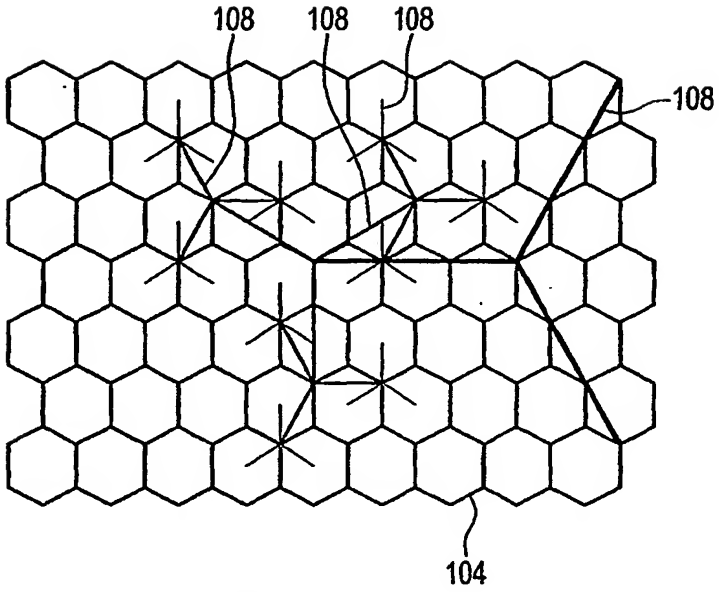


FIG. 39

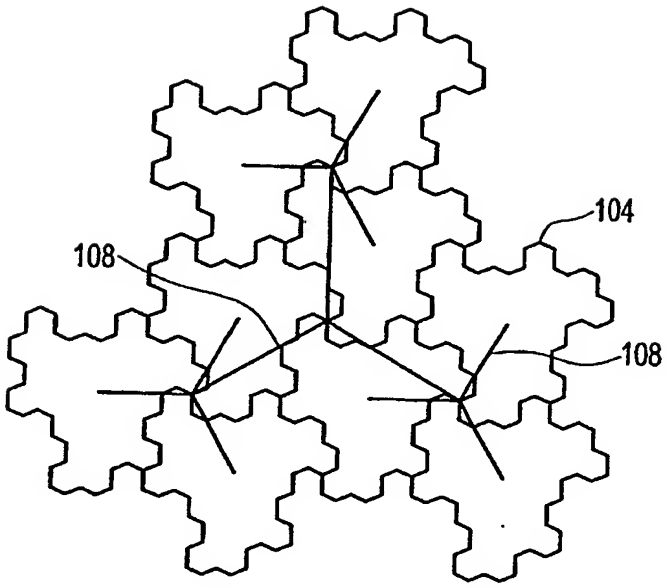


FIG. 40

Type	L
With dead cells	$L_1 = \sqrt{3}$ $L_n = 3L_{n-1} + 6^{n-1}\sqrt{3}$
Without dead cells	$L_1 = \sqrt{3}$ $L_n = 3L_{n-1} + 3^{n-1}\sqrt{3}^n$

Type	D
With dead cells	$D_1 = 2\sqrt{3}$ $D_n = 3D_{n-1} + 2\sqrt{3}(2^{n-1}-1)9^{n-1}$
Without dead cells	$D_1 = 2\sqrt{3}$ $D_n = 3D_{n-1} + (\sqrt{3}+3)(\sqrt{3}^n - 1)9^{n-1}$

FIG. 41

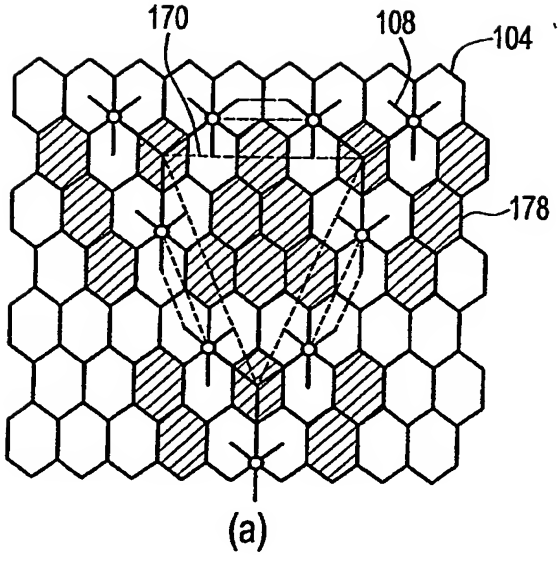


FIG. 42A

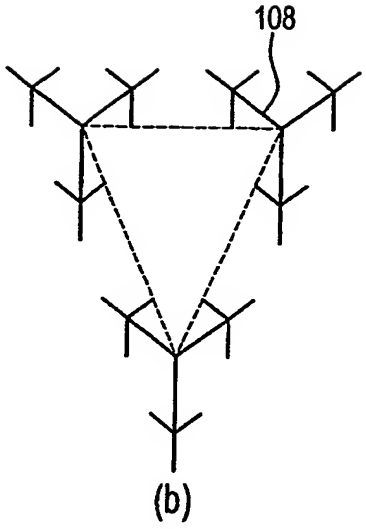


FIG. 42B

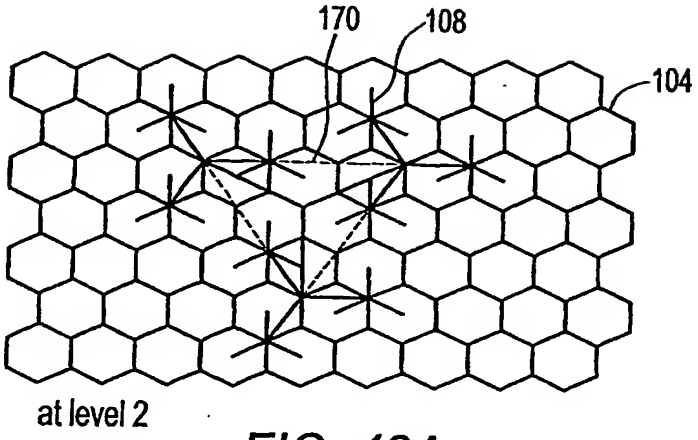


FIG. 43A

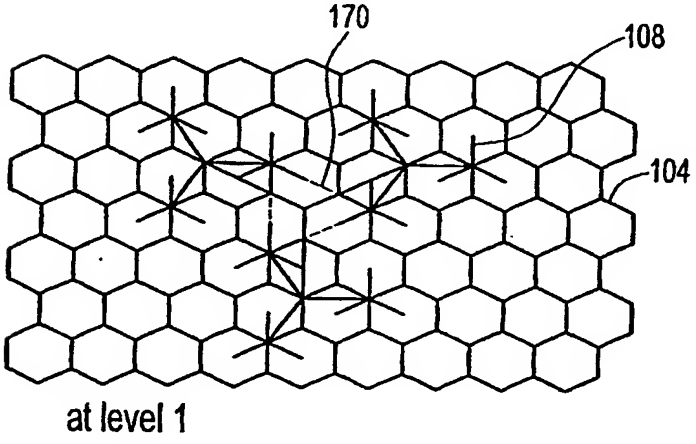


FIG. 43B

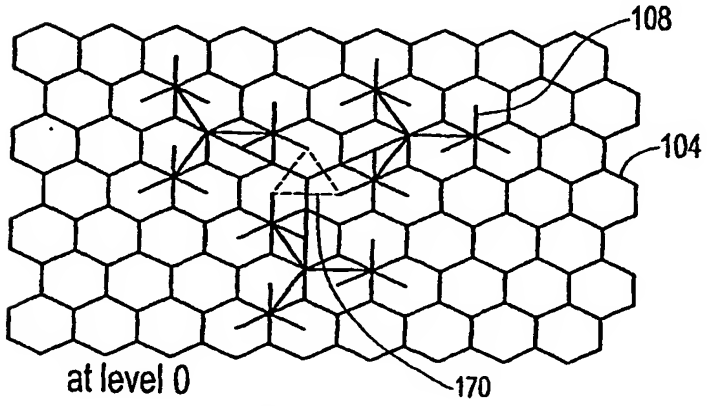


FIG. 43C

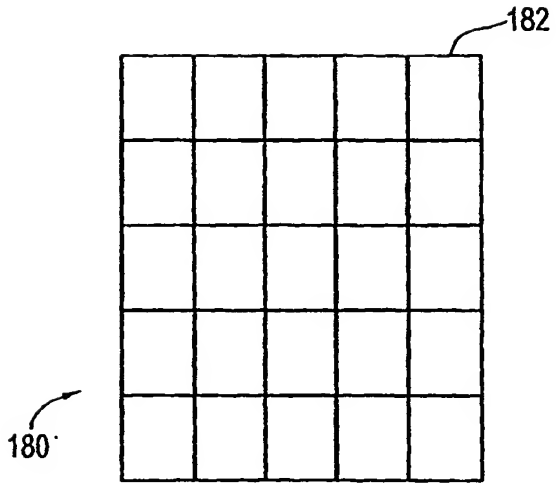


FIG. 44

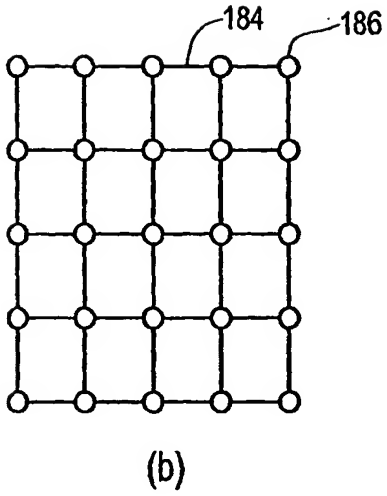
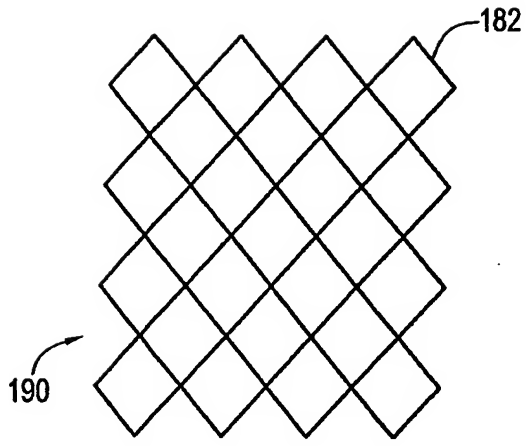
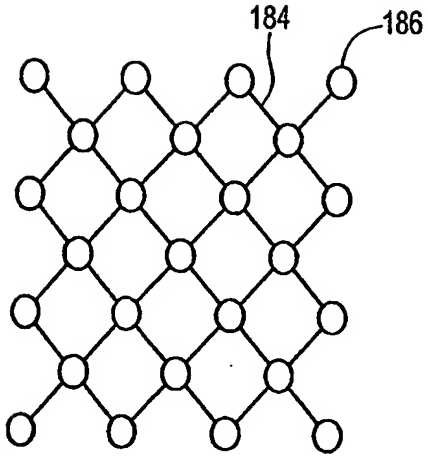


FIG. 45



(a)

FIG. 46



(b)

FIG. 47

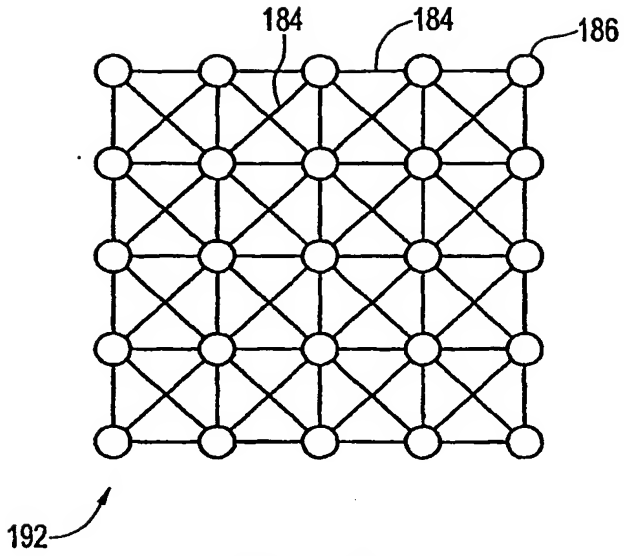
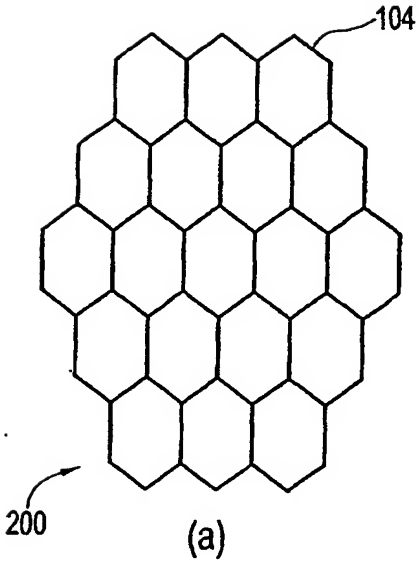
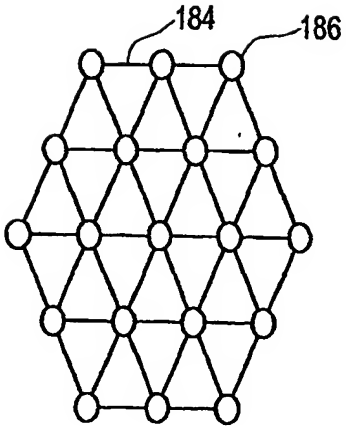


FIG. 48



(a)
he

FIG. 49



(b)
he

FIG. 50

Algorithm
For all $e \in E$, set $d_e = constant$
Repeat
For $j := 1$ **to** k **do** //k: number of distinct flow demands
Begin
Set $d(j) = \sigma$
While $d(j) \neq 0$ **do**
Begin
Find shortest path P for commodity flow demand j.
Route $f = min\{c, d(j)\}$ units of flow along P, where c is the capacity of the minimum capacity edge on this path.
$d(j) = d(j) - f$
Update $\{d_e\}$.
End while
End for
Find $\{C_1, C_2, ..., C_m\}$, such that $\sum_{e \in R(i)} d_e = \alpha_i$ and $\sum_i \alpha_i C_i = 1$
Update $\{d_e\}$
Until flow solutions converge

FIG. 51

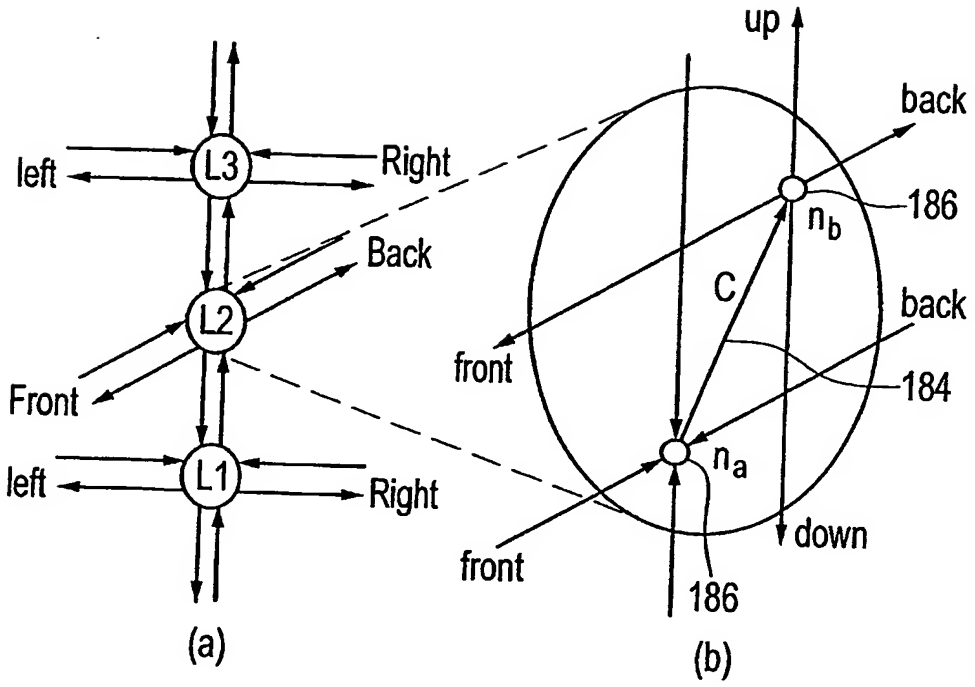

left
L3
Right
Back
L2
Front
left
L1
Right
(a)
up
back
186
n_b
C
back
184
front
n_a
down
front
186
(b)

FIG. 52

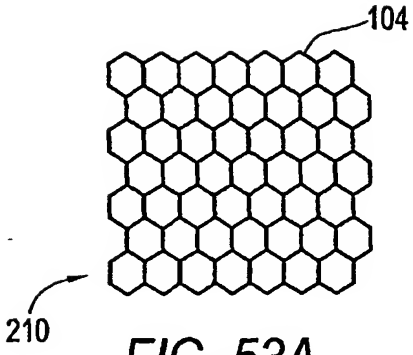


FIG. 53A

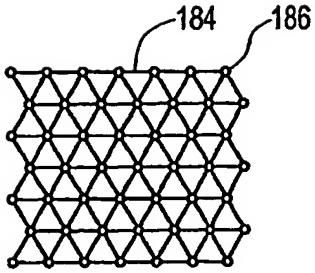


FIG. 53B

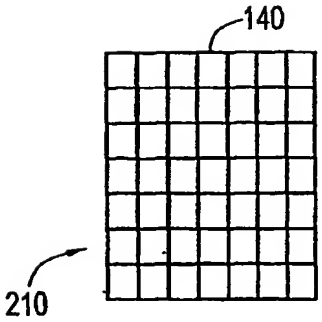


FIG. 54

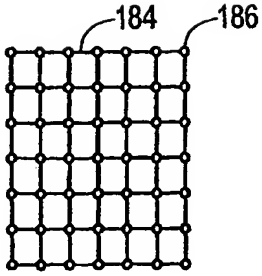


FIG. 55

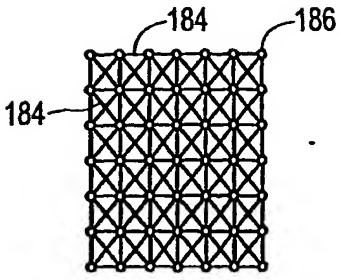


FIG. 56

Results of uniform edge capacity mesh

n	Number of nodes	z
2	4	0.3750
3	9	0.3333
4	16	0.2343
5	25	0.2000
6	36	0.1620
7	49	0.1429
8	64	0.1229
9	81	0.1111
10	100	0.0990

FIG. 57

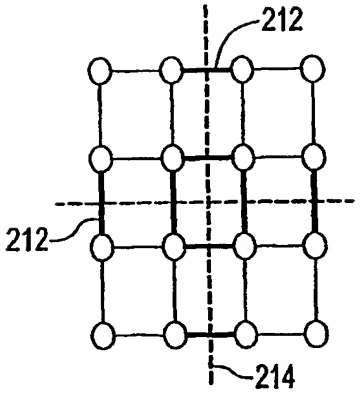


FIG. 58A

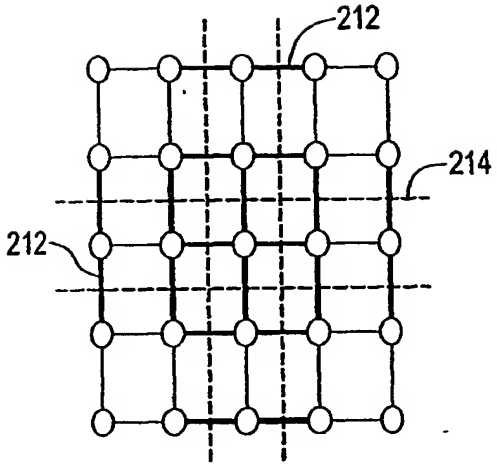


FIG. 58B

Results of fixed total edge capacities

N	Number of nodes	z	Improvement on z (%)
2	4	0.375	0.00
3	9	0.333	0.00
4	16	0.281	20.01
5	25	0.240	20.00
6	36	0.208	28.57
7	49	0.185	28.56
8	64	0.169	33.32
9	81	0.148	33.35
10	100	0.134	36.36

FIG. 59

Optimal capacities for vertical edges in 6 by 6 mesh

Row \ Col	1	2	3	4	5	6	Sum
1	0.60	0.74	0.79	0.79	0.74	0.61	4.28
2	0.95	1.19	1.27	1.28	1.19	0.96	6.85
3	1.07	1.34	1.44	1.44	1.34	1.07	7.71
4	0.95	1.19	1.27	1.27	1.19	0.96	6.85
5	0.60	0.74	0.79	0.79	0.74	0.60	4.28

FIG. 60

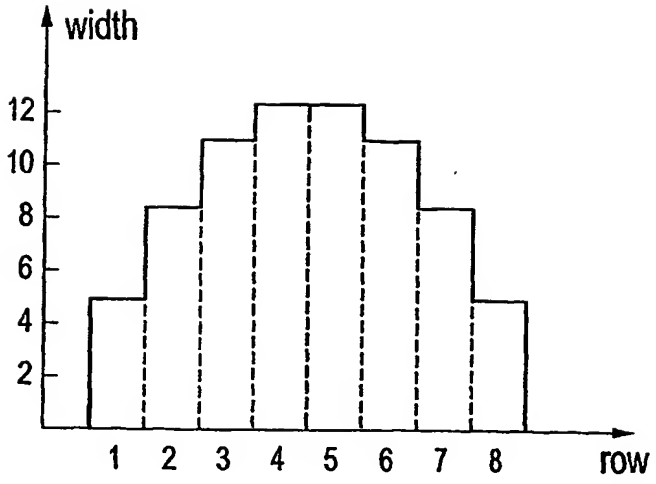


FIG. 61

Results of 45-degree mesh

N	Number of nodes	z
2	5	0.250
3	13	0.250
4	25	0.209
5	41	0.174
6	61	0.147
7	85	0.126
8	113	0.106
9	145	0.101
10	181	0.0828
11	221	0.0759
12	265	0.0673

FIG. 62

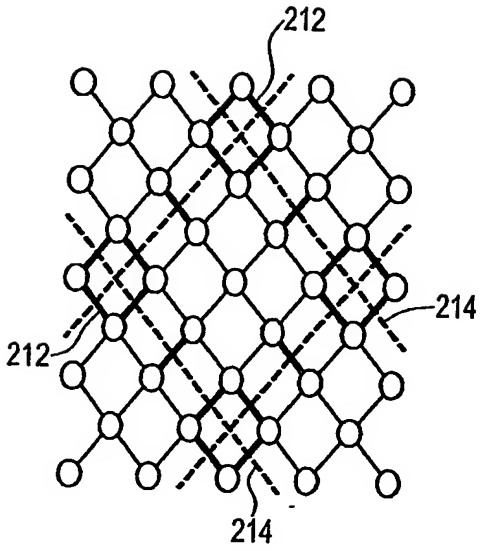


FIG. 63A

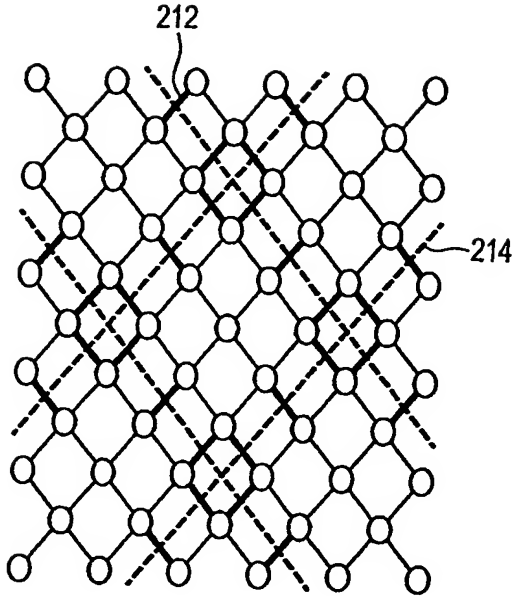


FIG. 63B

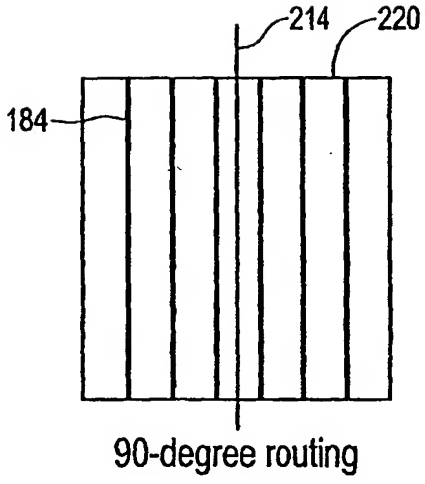


FIG. 64A

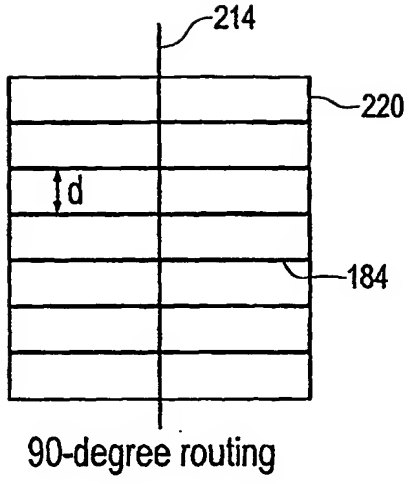
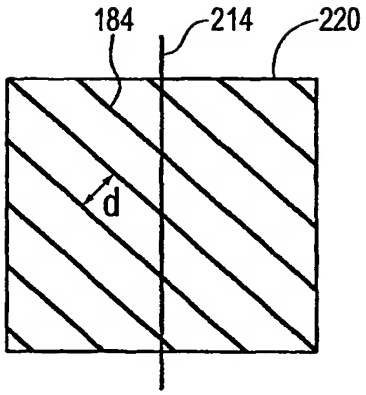
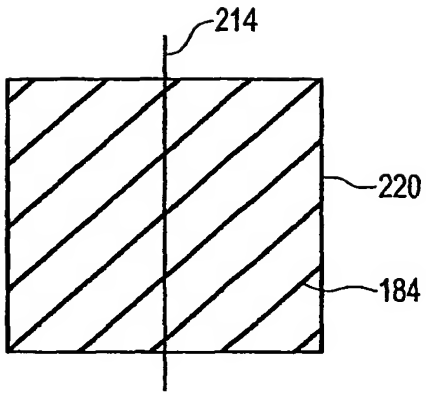


FIG. 64B



45-degree routing

FIG. 65A



45-degree routing

FIG. 65B

Results of 90-degree and 45-degree mixed mesh

n	z	Improvement on z (%)	C_1	C_2	$\sqrt{2} \cdot C_2 / C_1$
2	0.375	0.00	1.0000	0.0000	0.00
3	0.333	0.00	1.0000	0.0000	0.00
4	0.245	4.85	0.2290	0.5452	3.36
5	0.219	9.53	0.2577	0.5249	2.88
6	0.185	14.04	0.1853	0.5761	4.39
7	0.166	16.01	0.2022	0.5641	3.94
8	0.148	20.11	0.1614	0.5930	5.19
9	0.134	20.40	0.1696	0.5872	4.89
10	0.120	21.31	0.1553	0.5988	5.44
11	0.110	21.48	0.1608	0.5935	5.22
12	0.101	22.05	0.1527	0.5992	5.55
13	0.094	22.14	0.1562	0.5967	5.40
14	0.087	22.68	0.1510	0.6004	5.62
15	0.082	22.71	0.1536	0.5986	5.51
16	0.076	22.95	0.1504	0.6008	5.65
17	0.0723	23.02	0.1524	0.5994	5.56

FIG. 66

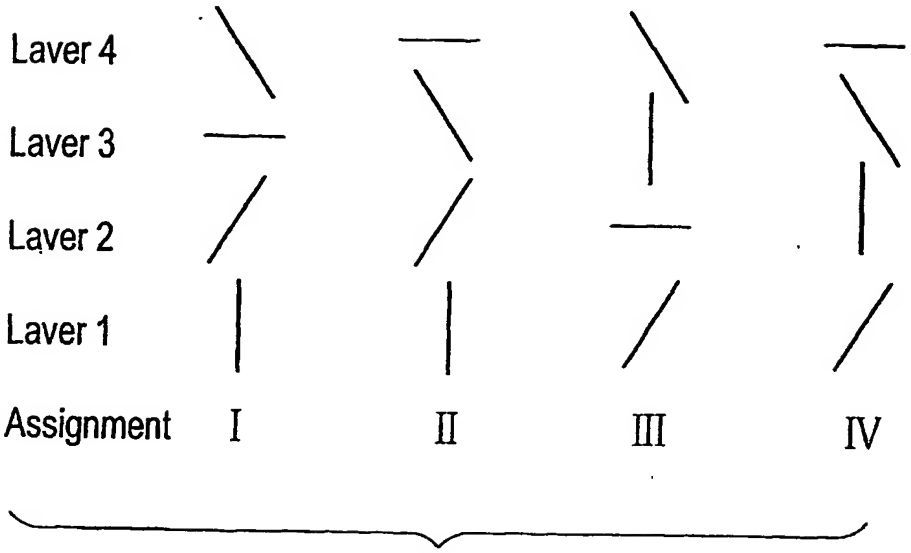


FIG. 67

N	z(I)	z(II)	z(III)	z(IV)
5	0.0173	0.0147	0.0147	0.0171
6	0.0102	0.0083	0.0083	0.0101
7	0.0065	0.0053	0.0051	0.0064
8	0.0041	0.0034	0.0034	0.0041

FIG. 68

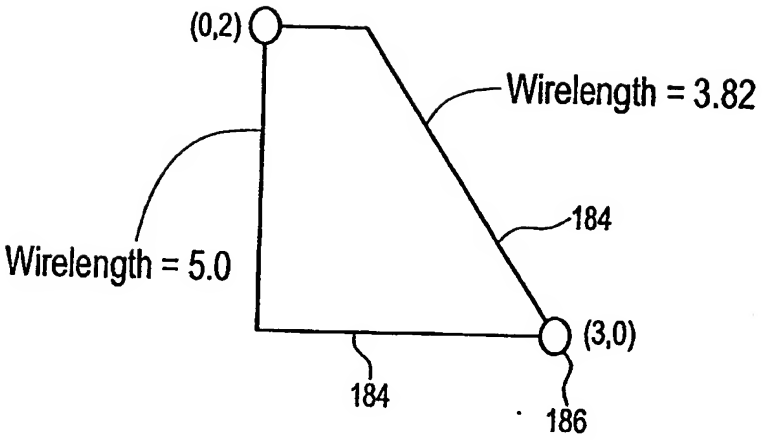


FIG. 69

n	#nodes	M-architecture		Y-architecture		X-architecture		
		thrpt	n.thrpt	n..thrpt	impr.(%)	n.thrpt	impr. (%)	$\sqrt{2}\cdot C_2/C_1$
2	4	2.50E-1	2.00	2.00	0	2.00	0	0.00
3	9	8.33E-2	2.25	2.25	0	2.25	0	0.00
4	16	3.12E-2	2.00	2.36	18.0	2.60	29.8	3.36
5	25	1.67E-2	2.09	2.40	20.1	2.68	28.1	2.88
6	36	9.26E-3	2.00	2.41	20.4	2.65	32.8	4.39
7	49	5.95E-3	2.04	2.41	20.4	2.67	31.1	3.94
8	64	3.90E-3	2.00	2.38	19.1	2.69	34.6	5.19
9	81	2.78E-3	2.03	2.45	22.5	2.69	32.7	4.89
10	100	1.98E-3	2.00	2.43	21.3	2.67	33.3	5.44
11	121	1.51E-3	2.01	2.46	23.1	2.70	34.4	5.12
12	144	1.16E-3	2.00	2.43	21.4	2.69	34.5	5.26
13	169	9.15E-4	2.01	2.43	21.5	2.70	34.4	5.33
14	196	7.29E-4	2.00	2.43	21.5	2.69	34.5	5.62
15	225	5.95E-4	2.01	2.43	21.6	2.70	34.5	5.51
16	256	4.88E-4	2.00	2.44	22.0	2.69	34.6	5.65
17	289	4.08E-4	2.00	2.45	22.5	2.70	34.6	5.56

FIG. 70

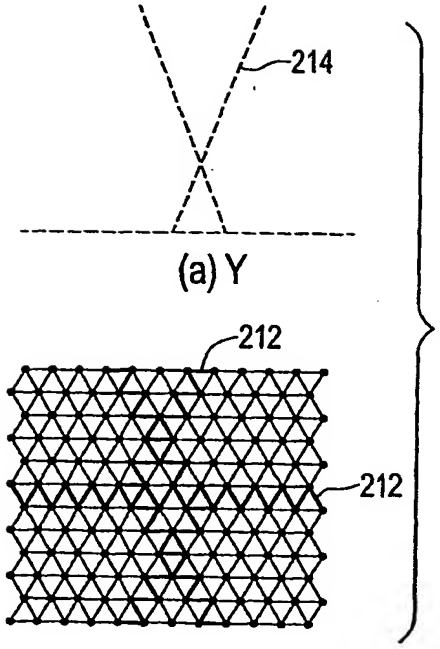


FIG. 71

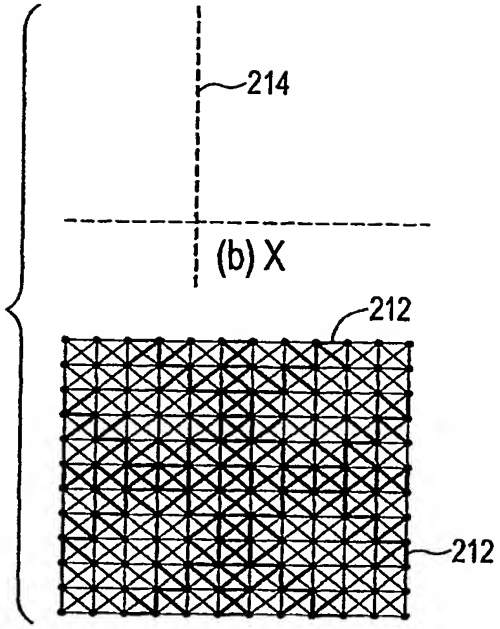


FIG. 72

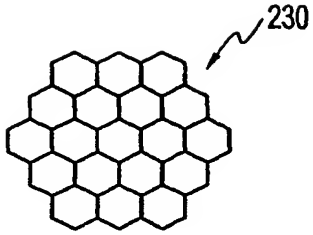


FIG. 73A

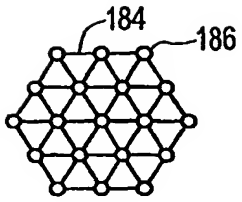


FIG. 73B

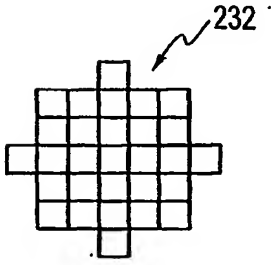


FIG. 73C

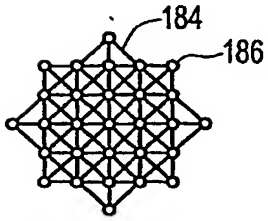


FIG. 73D

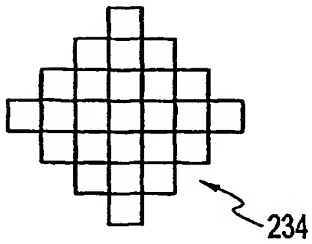


FIG. 73E

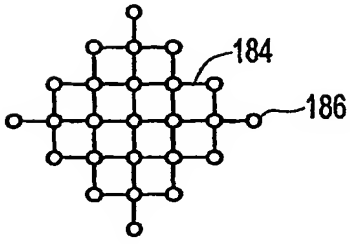


FIG. 73F

Level	#nodes	throughput	Normalized throughput
1	7	1.86E-1	2.02
2	19	1.69E-2	2.32
3	37	1.15E-3	2.48
4	61	5.33E-3	2.58
5	91	2.28E-3	2.61
6	127	4.41E-4	2.61
7	169	1.29E-4	2.62

FIG. 74

Level	#nodes	Throughput	Normalized throughput
2	29	2.31E-2	2.34
3	61	5.45E-3	2.51
4	101	3.01E-3	2.63
5	169	1.36E-3	2.74
6	281	5.75E-4	2.84

FIG. 75

Level	#nodes	Throughput	Normalized throughput
2	5	1.25E-1	1.78
3	13	4.20E-2	1.80
4	25	1.74E-2	2.09
5	41	8.71E-3	2.23
6	61	4.92E-3	2.30
7	85	3.00E-3	2.32
8	113	1.89E-3	2.36
9	145	1.39E-3	2.37
10	181	9.23E-4	2.38
11	221	6.90E-4	2.38
12	265	5.11E-4	2.39

FIG. 76

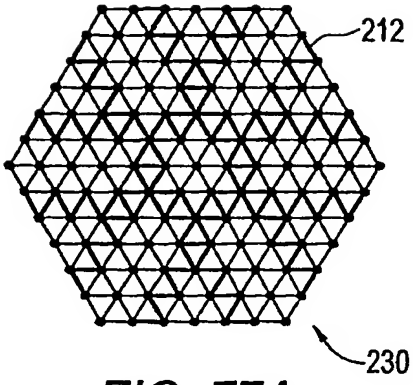


FIG. 77A

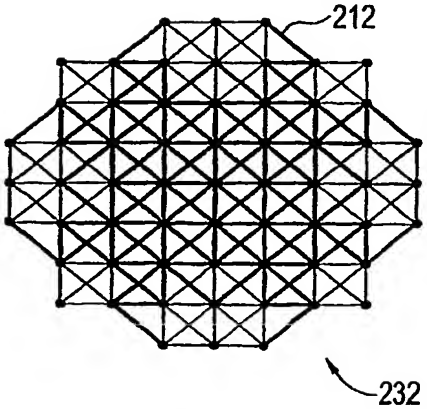


FIG. 77B

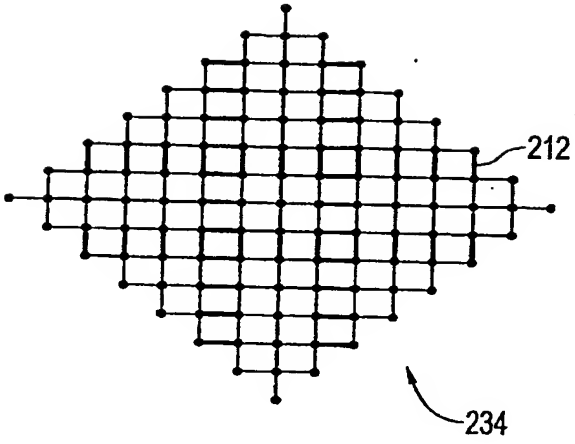


FIG. 77C